IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	Confirmation No. 5383
Li Gong	Art Unit: 2437
Serial No.: 08/883,636	Examiner: Callahan, Paul
Filed: June 26, 1997	Atty Docket No. P2145-US-NP
For: **LAYER-INDEPENDENT SECURITY FOR COMMUNICATION CHANNELS**	

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

I. REAL PARTY IN INTEREST

Oracle America, Inc.
500 Oracle Parkway
Redwood City, CA 94065
USA

II. RELATED APPEALS AND INTERFERENCES

No appeal or interference is known to Appellant that will affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-8, 13-20, 22-24, 26-32, 34 and 35 are pending in the application with claims 9-12, 21, 25 and 33 having been canceled. The rejection of claims 1-8, 13-20, 22-24, 26-32, 34 and 35 is the subject of this appeal.

IV. STATUS OF AMENDMENTS

No claim amendments were filed after the mailing of the Final Office Action on September 24, 2001 ("Final Office Action"). All other claim amendments have been entered, and pending appealed claims 1-8, 13-20, 22-24, 26-32, 34 and 35 are provided in the attached Claims Appendix.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 1:

Independent claim 1 is directed to "a method for providing communication protocol layer independent security for data transmitted between a first process (e.g., process 108, 208, see Appellant's Figures 1-2), executing on a first network node (e.g., node 102, 202), and a second process (e.g., process 110, 210), executing on a second network node (e.g., node 104, 204), wherein the first network node and the second network node each support at least one common communication protocol layer (e.g., socket layer 212, 214)."

The method of independent claim 1 includes "establishing a communication channel between the first network node and the second network node" (e.g., Java secure channel 216); "establishing a first stream between the first process and the communication channel" (e.g., Java output stream 218); and "establishing a second stream between the second process and the communication channel" (e.g., Java input stream 220). See steps 404-408 in Figure 4; page 9, lines 3-6; page 9, line 22 through page 10, line 11; and page 11, lines 4-11 of Appellant's specification.

The method of independent claim 1 then includes "in response to the data being written to the first stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., see step 412 in Figure 4; page 9, lines 16-18; page 11, lines 13-14; and page 12, lines 10-12). As the data encryption is performed in a manner that is independent of any communication protocol layers used to transport the encrypted data between the network nodes, the method of independent claim 1 is advantageously useful in object-oriented environments where there is an inherent level of abstractness required. See Appellant's specification at page 3, lines 10-11 and page 9, lines 16-

18. For instance, the encrypting may be performed above all communication protocol layers. See Appellant's page 12, lines 8-10.

The method of independent claim 1 also includes "causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes" (e.g., see step 416 in Figure 4) and "in response to the encrypted data being read from the second stream (e.g., step 418), decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream (e.g., step 420 and page 6, lines 20-22), the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., page 12, lines 16-18 and page 16, lines 6-8: "...data/stream [may] be decrypted at any layer..."). For example, an encrypted stream transmitted by a sending node may be decrypted by a firewall connection at the network packet layer having knowledge of the encryption approach negotiated during system setup. See page 16, lines 8-10.

Independent Claim 5:

Independent claim 5 is directed to "a computer-readable medium carrying one or more sequences of one or more instructions for providing communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol layer."

In independent claim 5, "the one or more sequences of one or more instructions [include] instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of: establishing a communication channel between the first network node and the second network node (e.g., Java secure channel 216); establishing a first stream between the first process and the communication channel (e.g., Java output stream 218); and establishing a second stream between the second process and the communication channel" (e.g., Java input stream 220). See steps 404-408 in Figure 4; page 9, lines 3-6; page 9, line 22 through page 10, line 11; and page 11, lines 4-11 of Appellant's specification.

The processors of independent claim 5 also include "in response to the data being written to the first stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., see step 412 in Figure 4; page 9, lines 16-18; page 11, lines 13-14; and page 12, lines 10-12). As the data encryption is performed in a manner that is independent of any communication protocol layers used to transport the encrypted data between the network nodes, the method of independent claim 5 is advantageously useful in object-oriented environments where there is an inherent level of abstractness required. See Appellant's specification at page 3, lines 10-11 and page 9, lines 16-18. For instance, the encrypting may be performed above all communication protocol layers. See Appellant's page 12, lines 8-10.

The processors of independent claim 5 also include "causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes" (e.g., see step 416 in Figure 4) and "in response to the encrypted data being read from the second stream (e.g., step 418), decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream (e.g., step 420 and page 6, lines 20-22), the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., page 12, lines 16-18 and page 16, lines 6-8: "...data/stream [may] be decrypted at any layer..."). For example, an encrypted stream transmitted by a sending node may be decrypted by a firewall connection at the network packet layer having knowledge of the encryption approach negotiated during system setup. See page 16, lines 8-10.

<u>Independent Claim 13:</u>

Independent claim 13 discloses "a computer data signal embodied in a carrier wave and representing sequences of instruction which, when executed by one or more processors, provide communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node,

according to at least one common communication protocol layer supported by the first and second network nodes."

The instructions of independent claim 13 include "establishing a communication channel between the first network node and the second network node" (e.g., Java secure channel 216); "establishing a first stream between the first process and the communication channel" (e.g., Java output stream 218); and "establishing a second stream between the second process and the communication channel" (e.g., Java input stream 220). See steps 404-408 in Figure 4; page 9, lines 3-6; page 9, line 22 through page 10, line 11; and page 11, lines 4-11 of Appellant's specification.

The instructions of independent claim 13 then include "in response to the data being written to the first stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., see step 412 in Figure 4; page 9, lines 16-18; page 11, lines 13-14; and page 12, lines 10-12). As the data encryption is performed in a manner that is independent of any communication protocol layers used to transport the encrypted data between the network nodes, the method of independent claim 13 is advantageously useful in object-oriented environments where there is an inherent level of abstractness required. See Appellant's specification at page 3, lines 10-11 and page 9, lines 16-18. For instance, the encrypting may be performed above all communication protocol layers. See Appellant's page 12, lines 8-10.

The instructions of independent claim 13 also includes "causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes" (e.g., see step 416 in Figure 4) and "in response to the encrypted data being read from the second stream (e.g., step 418), decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream (e.g., step 420 and page 6, lines 20-22), the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., page 12, lines 16-18 and page 16, lines 6-8: "...data/stream [may] be decrypted at any layer..."). For example, an encrypted stream transmitted by a sending

node may be decrypted by a firewall connection at the network packet layer having knowledge of the encryption approach negotiated during system setup. See page 16, lines 8-10.

Independent Claim 17:

Independent Claim 17 is directed to "a method for providing communication protocol layer independent security for data transmitted by a process executing on a network node," including "establishing a stream between the process and a communication channel" (e.g., Java output stream 218 in Figure 2; also see page 11, lines 4-11) and "in response to the data being written to the stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data on the communication channel." For example, see step 412 in Figure 4; page 9, lines 16-18; page 11, lines 13-14; and page 12, lines 10-12 of Appellant's specification. As the data encryption is performed in a manner that is independent of any communication protocol layers used to transport the encrypted data between the network nodes, the method of independent claim 17 is advantageously useful in object-oriented environments where there is an inherent level of abstractness required. See Appellant's specification at page 3, lines 10-11 and page 9, lines 16-18. For instance, the encrypting may be performed above all communication protocol layers. See Appellant's page 12, lines 8-10.

Independent Claim 20:

Independent claim 20 is directed to "a method for providing communication protocol-independent security for data transmitted between a first node and a second node." This method includes "establishing a communication channel between a first network node and a second network node" (e.g., Java secure channel 216 in Appellant's Figure 2; also see page 9, lines 3-6); "establishing a first stream from a first process to the communication channel after the establishment of the communication channel (e.g., Java output stream 218; also see page 11, lines 4-11), wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers (e.g., see step 412 in Figure 4 and page 9, lines 16-18); and "establishing a second stream from the communication channel to a second process after the establishment of the communication channel (e.g., Java input stream 220), wherein the second stream is decrypted

after the communication channel and before entering the second process" (see page 12, lines 3-7).

Independent Claim 24:

Independent claim 24 is directed to "a computer-readable medium carrying one or more sequences of one or more instructions for providing communication protocol-layer independent security for data transmitted between a first node and a second node, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

establishing a communication channel between a first network node and a second network node (e.g., Java secure channel 216 in Appellant's Figure 2; also see page 9, lines 3-6);

establishing a first stream from a first process to the communication channel after the establishment of the communication channel (e.g., Java output stream 218; also see page 11, lines 4-11), wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers (e.g., see step 412 in Figure 4 and page 9, lines 16-18); and

establishing a second stream from the communication channel to a second process after the establishing of the communication channel (e.g., Java input stream 220), wherein the second stream is decrypted after the communication channel and before entering the second process (see page 12, lines 3-7)."

Independent Claim 28:

Independent claim 28 is directed to "a communications network providing communication protocol-independent security for data transmitted between the first node and a second node, the communication network performing the steps of:

establishing a communication channel between a first network node and a second network node (e.g., Java secure channel 216 in Appellant's Figure 2; also see page 9, lines 3-6);

establishing a first stream from a first process to the communication channel after the establishment of the communication channel (e.g., Java output stream 218; also see page 11, lines 4-11), wherein the first stream is encrypted after the first process and before entering the

communication channel and the encrypted first stream is independent of any communication protocol layers (e.g., see step 412 in Figure 4 and page 9, lines 16-18); and

establishing a second stream from the communication channel to a second process after the establishing of the communication channel (e.g., Java input stream 220), wherein the second stream is decrypted after the communication channel and before entering the second process (see page 12, lines 3-7)."

Independent Claim 32:

Independent claim 32 is directed to "a computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by one or more processors, provide communication protocol-independent security for data transmitted between a first node and second node, by performing the steps of:

establishing a communication channel between a first network node and a second network node (e.g., Java secure channel 216 in Appellant's Figure 2; also see page 9, lines 3-6);

establishing a first stream from a first process to the communication channel after the establishment of the communication channel (e.g., Java output stream 218; also see page 11, lines 4-11), wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers (e.g., see step 412 in Figure 4 and page 9, lines 16-18); and

establishing a second stream from the communication channel to a second process after the establishing of the communication channel (e.g., Java input stream 220), wherein the second stream is decrypted after the communication channel and before entering the second process (see page 12, lines 3-7)."

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1) Whether claims 1, 5, 13, 17, 20, 24, 28 and 32 are unpatentable under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 5,793,749 to Helwig et al. ("Helwig").

2) Whether claims 1, 5, 13, 17, 20, 24, 28 and 32 are unpatentable under 35 U.S.C. §102(e) as being anticipated by "Applied Cryptography" to Schneier ("Schneier").

3) Whether claims 2, 3, 4, 6, 7, 8, 14, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31 and 34 are unpatentable under 35 U.S.C. §103(a) over either Helwig or Schneier.

VII. ARGUMENTS

Rejection of Claims 1, 5, 13, 17, 20, 24, 28, and 32 Under 35 U.S.C. §102(e) Based on Helwig Is Improper

In the Final Office Action, claims 1, 5, 13, 17, 20, 24, 28, and 32 were rejected under 35 U.S.C. §102(e) as being anticipated by Helwig. This rejection is traversed because Helwig does not disclose each and every limitation in claims 1, 5, 13, 17, 20, 24, 28, and 32.

In response to the Final Office Action, Appellant appealed the rejection such that there was no Advisory Action. The following discussion will address the Examiner's claim rejections and "Response to Arguments" section from the Final Office Action.

Independent Claims 1, 5 and 13:

Independent claims 1, 5 and 13 each include providing communication protocol layer independent security for data transmitted between a first process (e.g., process 108, 208, see Appellant's Figures 1-2), executing on a first network node (e.g., node 102, 202), and a second process (e.g., process 110, 210), executing on a second network node (e.g., node 104, 204), wherein the first network node and the second network node each support at least one common communication protocol layer (e.g., socket layer 212, 214).

Independent claims 1, 5 and 13 each include "establishing a communication channel between the first network node and the second network node" (e.g., Java secure channel 216); "establishing a first stream between the first process and the communication channel" (e.g., Java output stream 218); and "establishing a second stream between the second process and the communication channel" (e.g., Java input stream 220). See steps 404-408 in Figure 4; page 9, lines 3-6; page 9, line 22 through page 10, line 11; and page 11, lines 4-11 of Appellant's specification.

Independent claims 1, 5 and 13 each include "in response to the data being written to the first stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., see step 412 in Figure 4; page 9, lines 16-18; page 11, lines 13-14; and page 12, lines 10-12). As the data encryption is performed in a manner that is independent of any communication protocol layers used to transport the encrypted data between the network nodes, each of independent claims 1, 5 and 13

is advantageously useful in object-oriented environments where there is an inherent level of abstractness required. See Appellant's specification at page 3, lines 10-11 and page 9, lines 16-18. For instance, the encrypting may be performed above all communication protocol layers. See Appellant's page 12, lines 8-10.

Each of independent claims 1, 5 and 13 also includes "causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes" (e.g., see step 416 in Figure 4) and "in response to the encrypted data being read from the second stream (e.g., step 418), decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream (e.g., step 420 and page 6, lines 20-22), the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" (e.g., page 12, lines 16-18 and page 16, lines 6-8: "...data/stream [may] be decrypted at any layer..."). For example, an encrypted stream transmitted by a sending node may be decrypted by a firewall connection at the network packet layer having knowledge of the encryption approach negotiated during system setup. See page 16, lines 8-10.

For example, if independent claims 1, 5 and 13 were applied to a communication system which corresponded to the OSI reference model, communications would first be established between the first network node (e.g., node 102) and the second network node (node 104). The request for connection would come from the process 108 to the application layer and appropriately process through the layers until a connection is set up to node 104. Once that is done, a first stream, say, for example, an MPEG control channel stream is established between the first process 108 and the communications channel which begins at application layer 118. At the other end, a stream would be established between the application layer 128 of node 104 and the process 110 for the MPEG control channel data. As set forth in independent claims 1, 5 and 13, in response to data being written to the first stream [from process 108] the data is encrypted to generate encrypted data which is then applied to the application layer 118. As the encryption is performed independently of any of the layers of the communications protocol stack, independent claims 1, 5 and 13 are advantageously useful in object-oriented environments where there is an inherent level of abstractness required.

Helwig does not anticipate independent claims 1, 5 and 13 based on the following remarks:

> 1) Helwig does not disclose:
>
> "causing the encrypted data to be transmitted from the first network node to the second network node according to at least one communication protocol layer supported by the first and second network nodes"; and
>
> "in response to the encrypted data being read from the second stream [associated with a second node], decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream, the decrypting being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node"
>
> as required by each of independent claims 1, 5 and 13

In contrast to independent claims 1, 5 and 13 whereby data that has been encrypted in a communication protocol layer independent manner *at a first node is transmitted to a second node and then decrypted at the second node also in a communication protocol layer independent manner*, Helwig discloses a half-duplex communication node that can perform duplex testing involving transmitting and receiving messages at the same node without any disclosure regarding how a message or data is processed at a second node. More particularly, Helwig provides absolutely no disclosure of decrypting data at a second node in a communication protocol layer independent manner as called for in independent claims 1, 5 and 13. Due to at least this failure of Helwig to disclose limitations required by independent claims 1, 5 and 13, Appellant respectfully urges the Board to overturn this rejection and indicate these claims and their respective dependent claims allowable.

The background of Helwig at column 1 discusses that while "full-duplex" devices (those that can both send and receive messages simultaneously) are generally advantageous over half-duplex devices due to more natural communication and flexible signaling arrangements, they are so at the cost of increased processing power requirements. In this regard, Helwig discloses a half-duplex tester node 12 (see Figure 2) that can perform duplex testing by way of initially applying a pre-transmit process 68 to a real time message obtained from a microphone 18 (while not operating in half-duplex receive mode) and then recording the test message in memory 42.

See Helwig at column 4, line 57 through column 6, line 16 and process 68 in Figure 3. Thereafter, the same node 12 operates in a duplex mode in which the node 12 performs a final transmit process 90 of the test message recorded in memory 42 while simultaneously performing a receive process 88 of a signal received at antenna 30. See Helwig at column 6, line 17 through column 8, line 44 and process 64 in Figure 4.

At column 2, line 47 through column 3, line 21, Helwig generally discloses how the node 12 may transmit a signal to a unit under test node 14 which receives and simultaneously transmits the signal back to itself via lookback link 16. See Figure 1 of Helwig.

However, there is no discussion in Helwig regarding the node 12 performing the above-discussed final transmit process 90 of the pre-processed test message to the node 14 which is decrypted by the node 14, <u>much less</u> where the decrypting at the node 14 includes "decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream," and <u>much less</u> where "the decrypting being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node" as specifically recited in each of independent claims 1, 5 and 13. *Rather*, the majority of Helwig is dedicated to discussing the ability of the half-duplex tester node 12 to effectively function as a full-duplex node with little or no discussion of how the node 14 specifically receives and processes messages.

Data in independent claims 1, 5 and 13 is advantageously encrypted in a communication protocol layer independent manner before being transmitted from a first node to a second and then being decrypted also in a communication protocol layer independent manner which allows for the interaction of applications that implement encryption and decryption at different protocol layers (e.g., such as the interaction of Simply Key Management for Internet Protocols (SKIP) which is implemented at the network layer and Secure Sockets Layer (SSL) which is implemented at the socket layer). See Appellant's specification at page 3, lines 1-9 and page 16, lines 4-13.

As Helwig fails to disclose limitations required by independent claims 1, 5 and 13, Appellant respectfully urges the Board to overturn this rejection and indicate as allowable these claims and their respective dependent claims (i.e., claims 2-4, 6-8 and 14-16).

<u>2) Helwig fails to disclose the establishment of "a first stream between [a] first process and [a] communication channel" and "a second stream between [a] second process and the communication channel" in the manner recited in independent claims 1, 5 and 13 and consistent with Appellant's specification</u>

At page 4, lines 9-14 of Appellant's specification, a "stream" is specifically defined as follows:

> "In the context of the invention, a "stream" is an abstraction which refers to the transfer or "flow" of data, in any format, from a single source, to a single destination. A stream typically flows through a channel or connection between the sender and receiver, in contrast to data packets, which are typically individually addressed and which may be routed independently to multiple recipients. Hence, an application can write data to, or read data from, a stream without knowing the actual destination or source, respectively, of the data."

At the top of page 3 of the Final Office Action, the Examiner asserted:

> "Helwig…in the paragraph[es] spanning columns five and six, teach[es] encrypting a <u>data stream</u> and then formatting it to be compatible with a transmission protocol." (Emphasis added).

Paragraphs five and six of Helwig generally discuss performing the above-discussed "pre-transmit" process 68 on an acoustic signal received by microphone 18. Part of the pre-transmit process 68 involve a task 72 which converts the acoustic signal into an electronic signal and a task 74 which involves an A/D converter 20 which digitizes the electrical signal into a digital audio <u>electronic data stream</u> which is vocoded at task 76 and encrypted at task 78. Thus, it appears that the Examiner is equating the "electronic data stream" disclosed by Helwig to the "first stream" and "second stream" recited in independent claims 1, 5 and 13.

In relation to the "data stream" disclosed by Helwig, Helwig states the following at column 5, lines 46-53:

> "Next, a task 76 vocodes, or compresses, this digital data stream within DSP 22. Task 76 in the preferred embodiments uses conventional linear predictive coding (LPC) techniques to vocode the data stream so that substantially all the information content of the message

> may be represented using fewer data bits and may be transmitted using a smaller bandwidth than would be required to transmit the unvocoded data stream." (Emphasis added).

Thus, the "data streams" disclosed by Helwig are a series of bits that are output from a vocoder and used as a description of the data's particular physical format.

In contrast, each of the recited first and second "streams" is, as defined at page 4 of Appellant's specification (reproduced above), an abstraction which has properties beyond merely being a string of binary digits. A "stream," as would be understood by a skilled software practitioner, is defined in object oriented languages such as Java and has a whole set of associated properties which *distinguishes* it from simply an arbitrary string of binary 1's and 0's as is the "data stream" of Helwig. Here, it appears the Examiner improperly likened the "data stream" of Helwig to the "first stream" and "second stream" of independent claims 1, 5 and 13 based solely on the similarity of their names (i.e., both use the word "stream") and without determining whether the use of the word "stream" in Helwig is the same as the use of "stream" in independent claims 1, 5 and 13 (which it is not).

As Helwig fails to disclose first and second "streams" in the manner recited in independent claims 1, 5 and 13 and in a manner consistent with Appellant's specification, Appellant respectfully urges the Board to overturn this rejection and allow claims 1-8 and 13-16 due to this additional deficiency of Helwig.

Independent Claim 17:

Independent claim 17 is directed to a method for providing communication protocol layer independent security for data transmitted by a process executing on a network node. The method includes "establishing a stream between the process and a communication channel" (e.g., Java output stream 218 in Figure 2; also see page 11, lines 4-11) and "in response to the data being written to the stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data on the communication channel." For example, see step 412 in Figure 4; page 9, lines 16-18; page 11, lines 13-14; and page 12, lines 10-12 of Appellant's specification. As the data encryption is performed in a manner that is independent of any communication protocol

layers used to transport the encrypted data between the network nodes, the method of independent claim 17 is advantageously useful in object-oriented environments where there is an inherent level of abstractness required. See Appellant's specification at page 3, lines 10-11 and page 9, lines 16-18. For instance, the encrypting may be performed above all communication protocol layers. See Appellant's page 12, lines 8-10.

As discussed above in relation to independent claims 1, 5 and 13, Helwig does not at least disclose "establishing a stream between the process and a communication channel" as called for in independent claim 17 and in a manner consistent with Appellant's specification. Accordingly, Appellant respectfully urges the Board to overturn this rejection and allow claims 17-19 due to these deficiencies of Helwig.

Independent Claims 20, 24, 28 and 32:

Independent claims 20, 24, 28 and 32 each include "establishing a communication channel between a first network node and a second network node (e.g., Java secure channel 216 in Appellant's Figure 2; also see page 9, lines 3-6);" "establishing a first stream from a first process to the communication channel after the establishment of the communication channel (e.g., Java output stream 218; also see page 11, lines 4-11), wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers (e.g., see step 412 in Figure 4 and page 9, lines 16-18);" and "establishing a second stream from the communication channel to a second process after the establishing of the communication channel (e.g., Java input stream 220), wherein the second stream is decrypted after the communication channel and before entering the second process (see page 12, lines 3-7)."

As discussed above in relation to independent claims 1, 5 and 13, Helwig does not at least disclose the establishment of a communication channel and first and second streams from respective first and second processes to the channel as called for in independent claims 20, 24, 28 and 32 and in a manner consistent with Appellant's specification. Accordingly, Appellant respectfully urges the Board to overturn this rejection and allow claims 20, 22-24, 26-32, 34 and 35 due to these deficiencies of Helwig.

Rejection of Claims 1, 5, 13, 17, 20, 24, 28, and 32 Under 35 U.S.C. §102(e) Based on Schneier Is Improper

In the Final Office Action, claims 1, 5, 13, 17, 20, 24, 28, and 32 were rejected under 35 U.S.C. §102(e) as being anticipated by Schneier. This rejection is traversed because Schneier does not disclose each and every limitation in claims 1, 5, 13, 17, 20, 24, 28, and 32 as will be discussed below.

Independent Claims 1, 5 and 13:

1) Schneier does not disclose:

"establishing a communication channel between the first network node and the second network node";

"establishing a first stream between the first process and the communication channel"; and

"establishing a second stream between the second process and the communication channel"

as required by each of independent claims 1, 5 and 13

Schneier discloses a high-level discussion of a "stream cipher" which involves encrypting plaintext to ciphertext one bit at a time using a keystream generator and then, at some later time, decrypting the ciphertext bits one bit at a time to recover the plaintext bits.

At page 3 of the Final Office Action, the Examiner merely stated "Claims 1, 5, 13, 17, 20, 24, 28 and 32 are rejected…as being clearly anticipated by Schneier…See figure 9.6." No further explanation or attempt to relate portions of Schneier to the recited claim limitations was provided. Furthermore, Appellant is unable to locate any portion of Schneier that is equivalent to the recited steps of "establishing a communication channel…a first stream…and…a second stream" as called for in independent claims 1, 5 and 13.

It appears the Examiner improperly likened the "stream cipher" of Helwig to the "first stream" and "second stream" of independent claims 1, 5 and 13 based solely on the similarity of their names (i.e., both use the word "stream") and without determining whether the use of the word "stream" in Helwig is the same as the use of "stream" in independent claims 1, 5 and 13 (which it is not). Again, each of the recited first and second "streams" is, as defined above at page 4 of Appellant's specification, an abstraction which, as would be understood by a skilled software practitioner, is defined in object oriented languages such as Java and has a whole set of

associated properties which *distinguishes* it from simply an arbitrary string of binary 1's and 0's as is the "stream cipher" of Schneier.

As Schneier fails to disclose establishment of a communication channel between first and second nodes along with establishment of first and second streams in the manner recited in independent claims 1, 5 and 13 and in a manner consistent with Appellant's specification, Appellant respectfully urges the Board to overturn this rejection and allow claims 1-8 and 13-16.

> <u>2) Schneier does not disclose:</u>
>
> <u>"encrypting ...data...independent of any communication protocol layers..."</u>
>
> <u>"causing the encrypted data to be transmitted from the first network node to the second network node according to at least one communication protocol layer supported by the first and second network nodes"; and</u>
>
> <u>"in response to the encrypted data being read from the second stream [associated with a second node], decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream, the decrypting being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node"</u>
>
> <u>as required by each of independent claims 1, 5 and 13</u>

As discussed above, Schneier merely provides a high-level discussion of a "stream cipher" which involves encrypting plaintext to ciphertext one bit at a time using a keystream generator and then, at some later time, decrypting the ciphertext bits one bit at a time to recover the plaintext bits. Furthermore, Appellant is unable to identify any portion of Schneier that would be equivalent to the above-recited steps of encrypting data in a communication layer protocol independent manner, causing the data to be transmitted from a first node to a second node, and decrypting the data at the second node in a communication protocol layer independent manner as called for in independent claims 1, 5 and 13. Accordingly, Appellant respectfully requests that the Board indicates claims 1-8 and 13-16 allowable based at least on these additional deficiencies of Schneier.

<u>Independent Claim 17:</u>

Independent claim 17 is directed to a method for providing communication protocol layer independent security for data transmitted by a process executing on a network node. The method

includes "establishing a stream between the process and a communication channel" (e.g., Java output stream 218 in Figure 2; also see page 11, lines 4-11) and "in response to the data being written to the stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data on the communication channel." For example, see step 412 in Figure 4; page 9, lines 16-18; page 11, lines 13-14; and page 12, lines 10-12 of Appellant's specification. As the data encryption is performed in a manner that is independent of any communication protocol layers used to transport the encrypted data between the network nodes, the method of independent claim 17 is advantageously useful in object-oriented environments where there is an inherent level of abstractness required. See Appellant's specification at page 3, lines 10-11 and page 9, lines 16-18. For instance, the encrypting may be performed above all communication protocol layers. See Appellant's page 12, lines 8-10.

As discussed above in relation to independent claims 1, 5 and 13, Schneier does not at least disclose "establishing a stream between the process and a communication channel" as called for in independent claim 17 and in a manner consistent with Appellant's specification. Accordingly, Appellant respectfully urges the Board to overturn this rejection and allow claims 17-19 due to these deficiencies of Schneier.

Independent Claims 20, 24, 28 and 32:

Independent claims 20, 24, 28 and 32 each include "establishing a communication channel between a first network node and a second network node (e.g., Java secure channel 216 in Appellant's Figure 2; also see page 9, lines 3-6);" "establishing a first stream from a first process to the communication channel after the establishment of the communication channel (e.g., Java output stream 218; also see page 11, lines 4-11), wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers (e.g., see step 412 in Figure 4 and page 9, lines 16-18);" and "establishing a second stream from the communication channel to a second process after the establishing of the communication channel (e.g., Java input stream 220), wherein the second stream is decrypted after the communication channel and before entering the second process (see page 12, lines 3-7)."

As discussed above in relation to independent claims 1, 5 and 13, Schneier does not at least disclose the establishment of a communication channel and first and second streams from respective first and second processes to the channel as called for in independent claims 20, 24, 28 and 32 and in a manner consistent with Appellant's specification. Accordingly, Appellant respectfully urges the Board to overturn this rejection and allow claims 20, 22-24, 26-32, 34 and 35 due to these deficiencies of Schneier.

Rejection of Claims 3, 4, 7, 8, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31 and 34 Under 35 U.S.C. §103(a) Based on Helwig or Schneier Is Improper

In the Final Office Action, claims 3, 4, 7, 8, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31 and 34 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schneier or Helwig. This rejection is traversed because Schneier and Helwig do not disclose or suggest each and every limitation in claims 3, 4, 7, 8, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31 and 34 as will be discussed below.

Initially, each of claims 3, 4, 7, 8, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31 and 34 depends from one of independent claims 1, 5, 13, 17, 20, 24, 28 or 32 which have already been shown to be allowable as discussed above. Thus, claims 3, 4, 7, 8, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31 and 34 are allowable for at least those reasons in support of the allowability of independent claims 1, 5, 13, 17, 20, 24, 28 and 32.

Furthermore, each of claims 3, 4, 7, 8, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31, 34 and 35 is not obvious over either Helwig or Schneier and is therefore allowable independently of depending from an allowable independent claim as discussed above because Helwing and Schneier both fail to teach or suggest a Java-based stream or communication channel as called for in these claims.

At pages 3-4 of the Final Office Action, the Examiner stated:

> "[Schneier and Helwig] do not say that the communication channels or data streams are Java-based. Official notice is taken that it is old and well-known that Java is intended for networked/distributed environments and enables the construction of virus-free, tamper-free systems. Therefore it would have been obvious to a person of ordinary skill in the art at the time the invention was made to base the systems of Schneier or Helwig et al., all of which are

> networked or distributed environments, on Java, as is known in the art. This would enable the implementation of a virus-free, tamper-free system."

Appellant continues to insist that the range and content of the Examiner's Official Notice is factually and legally erroneous. See Petition to the Group Director filed on April 11, 2001, the Response filed on May 14, 2001, and the Petition for Review of Director's Decision filed on July 19, 2001.

Even assuming for the sake of argument that the Official Notice was effective for what the Examiner asserts, Appellant urges that the requirements of 35 USC §103(a) have still not been satisfied. More specifically, the Examiner has not provided a cogent explanation of why one of ordinary skill would have been motivated to modify the message storing device of Helwig to add, for example, the complexity, additional hardware and cost of Java processing capability in the first place. Similarly, the Examiner has not explained why an ordinary artisan would have been motivated to augment the general discussion of enciphering and deciphering models by Schneier to specifically involve Java and Java streams. The Examiner's rationale that Java "enables construction of virus-free, taper-free systems" is exactly the danger of which the courts have repeatedly warned against and the type of reasoning which the courts have repeatedly found erroneous

Still further, even assuming the Examiner implemented the systems of Helwig or Schneier with Java streams and Java secure channels, such a modification would still not result in the claimed invention. More specifically, if the phrases "communication channel" and "stream" as used in Helwig and Schneier were interpreted to be "Java secure communication channel" and "Java stream," respectively, the interpretation of Helwig and Schneier would be required to change to an extent that would further support Appellant's position that these references are inapplicable as part of the rejections under 35 U.S.C. §102 presented herein.

Accordingly, Appellant respectfully urges the Board to overturn this rejection and allow claims 3, 4, 7, 8, 15, 16, 18, 19, 22, 23, 26, 27, 30, 31, 34 and 35 due to these deficiencies of Helwig and Schneier.

Rejection of Claims 2, 6 and 14 Under 35 U.S.C. §103(a) Based on Helwig or Schneier Is Improper

In the Final Office Action, claims 2, 6 and 14 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schneier or Helwig. This rejection is traversed because Schneier and Helwig do not disclose or suggest each and every limitation in claims 2, 6 and 14 as will be discussed below. Furthermore, while not specifically mentioned by the Examiner, Appellant will assume the Examiner intended to reject claim 29 under this section due to the similarity of subject matter between claim 29 and claims 2, 6 and 14.

Initially, each of claims 2, 6, 14 and 29 depends from one of independent claims 1, 5, 13 and 28 which have already been shown to be allowable as discussed above. Thus, claims 2, 6, 14 and 29 are allowable for at least those reasons in support of the allowability of independent claims 1, 5, 13 and 28.

Furthermore, each of claims 2, 6, 14 and 29 is not obvious over either Helwig or Schneier and is therefore allowable independently of depending from an allowable independent claim as discussed above because Helwing and Schneier both fail to teach or suggest communication protocol layer specific encryption and decryption in the manner recited in claims 2, 6, 14 and 29.

At page 4 of the Final Office Action, the Examiner again admitted that Helwig and Schneier fail to disclose communication protocol layer specific encryption and decryption. After again taking Official Notice (this time that it is well known to encrypt already encrypted data which is sometimes performed in communications protocols), the Examiner contended that it would have been obvious to encrypt the already encrypted data at a communication protocol layer to increase security. Appellant respectfully disagrees with the Examiner's combination of Helwig or Schneier with the Official Notice.

In rejecting these claims, the Examiner has essentially asserted that if some encryption is good, then more encryption is better. However, claims 2, 6, 14 and 29 require more than merely "additional encryption." More specifically, these claims require that the encryption is a "communication protocol layer specific" encryption. The Examiner has not explained why a skilled artisan would have found it obvious to add protocol specific encryption to the systems of Schneier or Helwig.

Schneier does not disclose a stream in the context of networked nodes communicating over a channel, and Helwig is concerned about storing a message as opposed to secure communications. Additionally, Helwig discusses the need for responsiveness in its system and one skilled in the art would not have adversely impacted performance in such a system by adding another layer of encryption processing. As the Examiner has thus failed to provide a fact-based rationale as to why one of ordinary skill would have been motivated to modify Schneier or Helwig with a second encryption/decryption step and why that skilled artisan would have performed the encryption/decryption in a protocol layer specific manner, Appellant respectfully urges the Board to overturn this rejection and allow claims 2, 6, 14 and 29 for this additional reason.

Conclusion

In view of the above remarks, the pending claims are believed allowable and the case is in condition for allowance. Appellant respectfully requests that the rejections of all pending claims be reversed. Appellant has met or exceeded the burden of overcoming the prima facie case of unpatentability made out by the Examiner by providing rebuttal evidence of adequate weight and explaining how and why the record indicates the pending claims are patentable.

Respectfully submitted,

Date: October 28, 2011

Jonathon A. Szumny, Reg. No. 57,695
Marsh Fischmann & Breyfogle LLP
8055 E. Tufts Ave., Suite 450
Denver, CO 80237
Phone: (303) 770-0051
Fax: (303) 770-0152

VIII. CLAIMS APPENDIX

1. A method for providing communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol layer, the method comprising the steps of:

establishing a communication channel between the first network node and the second network node;

establishing a first stream between the first process and the communication channel;

establishing a second stream between the second process and the communication channel;

in response to the data being written to the first stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node;

causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes; and

in response to the encrypted data being read from the second stream, decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream, the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node.

2. The method of Claim 1, further including the steps of

performing a communication protocol layer specific encryption of data to be sent across the communication channel at the first network node, and

performing a communication protocol layer specific decryption of data received from the communication channel at the second network node.

3. The method of Claim 1, wherein the communication channel is a Java secure channel,

wherein the first stream is a first Java stream,

wherein the second stream is a second Java stream,

wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel, and

wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel.

4. The method of Claim 1, wherein the communication channel is a Java secure channel, wherein the first stream is a Java stream,

wherein the second stream is a Java stream,

wherein the method further comprises the step of connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol layer.

5. A computer-readable medium carrying one or more sequences of one or more instructions for providing communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol layer, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

establishing a communication channel between the first network node and the second network node;

establishing a first stream between the first process and the communication channel;

establishing a second stream between the second process and the communication channel;

in response to the data being written to the first stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication

protocol layers used to transport the encrypted data from the first network node to the second network node;

causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes; and

in response to the encrypted data being read from the second stream, decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream, the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node.

6. The computer-readable medium of Claim 5, wherein the computer-readable medium further includes instructions for performing the steps of

performing a communication protocol layer specific encryption of the data on the first network node, and

performing a communication protocol layer specific decryption of the data on the second network node.

7. The computer-readable medium of Claim 5, wherein the first stream is a first Java stream,

wherein the second stream is a second Java stream,

wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel, and

wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel.

8. The computer-readable medium of Claim 5, wherein the communication channel is a Java secure channel,

wherein the first stream is a Java stream,

wherein the second stream is a Java stream,

wherein the computer-readable medium further includes instructions for connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol layer.

13. A computer data signal embodied in a carrier wave and representing sequences of instruction which, when executed by one or more processors, provide communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, according to at least one common communication protocol layer supported by the first and second network nodes, by performing the steps of:

establishing a communication channel between the first network node and the second network node;

establishing a fist stream between the first process and the communication channel;

establishing a second stream between the second process and the communication channel;

in response to the data being written to the first stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node;

causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes; and

in response to the encrypted data being read from the second stream, decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream, the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node.

14. The computer data signal of Claim 13, wherein the computer sequence of instructions further includes instructions for performing the steps of

performing a communication protocol layer specific encryption of the data on the first network node, and

performing a communication protocol layer specific decryption of the data on the second network node.

15. The computer data signal of Claim 13, wherein the first stream is a first Java stream,

wherein the second stream is a second Java stream,

wherein the step of establishing a communication channel between the first and second network nodes further comprises the step of establishing a Java secure channel between the first and second network nodes,

wherein the step of establishing a first stream between the first process and the communication channel further comprises the step of establishing a first Java stream between the first process and the Java secure channel,

wherein the step of establishing a second stream between the second process and the communication channel further comprises the step of establishing a second Java stream between the second process and the Java secure channel.

16. The computer data signal of Claim 13, wherein the communication channel is a Java secure channel,

wherein the first stream is a Java stream,

wherein the second stream is a Java stream,

wherein the computer sequence of instructions further includes instructions for connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol layer.

17. A method for providing communication protocol layer independent security for data transmitted by a process executing on a network node, the method comprising the steps of:

a) establishing a stream between the process and a communication channel; and

b) in response to the data being written to the stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data on the communication channel.

18. The method of Claim 17, wherein the communication channel is a Java secure channel, wherein the stream is a first Java stream, and

wherein the step of establishing a stream between the process and the communication channel further comprises the step of establishing a Java stream between the process and the Java secure channel.

19. The method of Claim 17, wherein the communication channel is a Java secure channel, wherein the stream is a Java stream,

wherein the method further comprises the step of connecting the Java secure channel to a second Java stream, and

wherein the second Java stream provides for the transmission of data according to a specific communication protocol layer.

20. A method for providing communication protocol-independent security for data transmitted between a first node and a second node, the method comprising the steps of:

establishing a communication channel between a first network node and a second network node;

establishing a first stream from a first process to the communication channel after the establishment of the communication channel, wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers; and

establishing a second stream from the communication channel to a second process after the establishment of the communication channel, wherein the second stream is decrypted after the communication channel and before entering the second process.

22. The method of claim 20, wherein:

the first stream is a first Java stream;

the second stream is a second Java stream;

the step of establishing a communication channel between the first network node and second network node further comprises the step of establishing a Java secure channel between the first network node and second network node;

the step of establishing the first stream comprises the step of establishing the first Java stream after the first process and before the Java secure channel; and

the step of establishing a second stream comprises the step of establishing the second Java stream after the Java secure channel and before the second process.

23. The method of clam 20, wherein;

the communication channel is a Java secure channel;

the first stream is a Java stream;

the second stream is a Java stream;

the method further comprises the step of connecting the Java secure channel to a third Java stream; and

the third Java stream provides for the transmission of data according to a specific communication protocol layer.

24. A computer-readable medium carrying one or more sequences of one or more instructions for providing communication protocol-layer independent security for data transmitted between a first node and a second node, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

establishing a communication channel between a first network node and a second network node;

establishing a first stream from a first process to the communication channel after the establishment of the communication channel, wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers; and

establishing a second stream from the communication channel to a second process after the establishing of the communication channel, wherein the second stream is decrypted after the communication channel and before entering the second process.

26. The computer-readable medium of claim 24, wherein:

the first stream is a first Java stream;

the second stream is a second Java stream;

the step of establishing a communication channel between the first network node and second network node further comprises the step of establishing a Java secure channel between the first network node and second network node;

the step of establishing the first stream comprises the step of establishing the first Java stream after the first process and before the Java secure channel; and

the step of establishing a second stream comprises the step of establishing the second Java stream after the Java secure channel and before the second process.

27. The method of claim 24, wherein:

the communication channel is a Java secure channel;

the first stream is a Java stream;

the second stream is a Java stream

the method further comprises the step of connecting the Java secure channel to a third Java stream; and

the third Java stream provides for the transmission of data according to a specific communication protocol layer.

28. A communications network providing communication protocol-independent security for data transmitted between the first node and a second node, the communication network performing the steps of:

establishing a communication channel between a first network node and a second network node;

establishing a first stream from a first process to the communication channel after the establishment of the communication channel, wherein the first stream is encrypted after the first

process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers; and

establishing a second stream from the communication channel to a second process after the establishment of the communication channel, wherein the second stream is decrypted after the communication channel and before entering the second process.

29. The communication network of claim 28, wherein the encryption of the first stream and the decryption of the second stream is specific to a communication protocol layer.

30. The communication network of claim 28, wherein:

the first stream is a first Java stream;

the second stream is a second Java stream;

the step of establishing a communication channel between the first network node and second network node further comprises the step of establishing a Java secure channel between the first network node and second network node;

the step of establishing the first stream comprises the step of establishing the first Java stream after the first process and before the Java secure channel; and

the step of establishing a second stream comprises the step of establishing the second Java stream after the Java secure channel and before the second process.

31. The communication network of claim 28, wherein:

the communication channel is a Java secure channel;

the first stream is a Java stream;

the second stream is a Java stream

the method further comprises the step of connecting the Java secure channel to a third Java stream; and

the third Java stream provides for the transmission of data according to a specific communication protocol layer.

32. A computer data signal embodied in a carrier wave and representing sequences of instructions which, when executed by one or more processor, provide communication protocol-

independent security for data transmitted between a first node and second node, by performing the steps of:

establishing a communication channel between a first network node and a second network node;

establishing a first stream from a first process to the communication channel after the establishment of the communication channel, wherein the first stream is encrypted after the first process and before entering the communication channel and the encrypted first stream is independent of any communication protocol layers; and

establishing a second stream from the communication channel to a second process after the establishment of the communication channel, wherein the second stream is decrypted after the communication channel and before entering the second process.

34. The computer data signal of claim 32, wherein:

the first stream is a first Java stream;

the second stream is a second Java stream;

the step of establishing a communication channel between the first network node and second network node further comprises the step of establishing a Java secure channel between the first network node and second network node;

the step of establishing the first stream comprises the step of establishing the first Java stream after the first process and before the Java secure channel; and

the step of establishing a second stream comprises the step of establishing the second Java stream after the Java secure channel and before the second process.

35. The computer data signal of claim 32, wherein:

the communication channel is a Java secure channel;

the first stream is a Java stream;

the second stream is a Java stream

the method further comprises the step of connecting the Java secure channel to a third Java stream; and

the third Java stream provides for the transmission of data according to a specific communication protocol layer.

IX. EVIDENCE APPENDIX

No copies of evidence are required with this Appeal Brief. Appellant has not relied upon any evidence submitted under 37 C.F.R. §§ 1.130, 1.131, or 1.132.

X. RELATED PROCEEDINGS APPENDIX

There are no copies of decisions rendered by a court or the Board to provide with this Appeal as there are no related proceedings.